TSG's FXClust Scalar

Peter Mason, October 2017

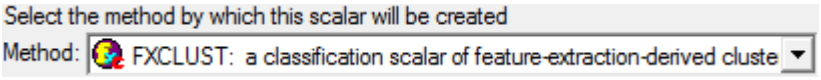
Introduction

In October 2017 a new scalar calculation "FXCLUST" was put into TSG. It implements a spectral clustering procedure developed by Andrew Rodger of CSIRO. It classifies dataset samples according to the grouping of their dominant spectral features. Here's an overview of how it works:

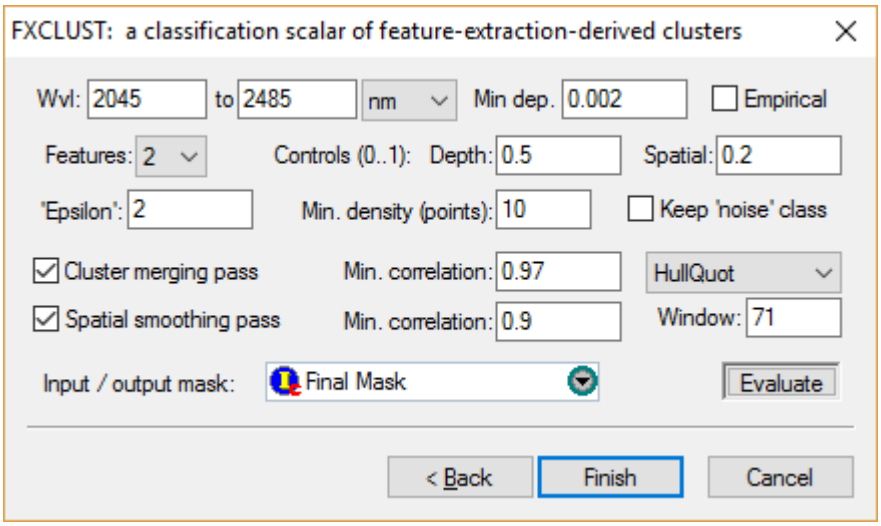
- It is driven by feature-extraction results.
    - Feature selection can be moderated by wavelength subsetting and a minimum depth threshold.
    - The wavelength centres of each sample's deepest 2 to 5 features are the main clustering inputs.
    - Two other experimental inputs may be included for each sample.
- The "DBSCAN" clustering algorithm is used to cluster the inputs.
    - Samples with similar features are clustered together.
    - The algorithm can track shaped clusters (not just blobs) like ones often seen in feature-extraction scatterplots.
    - The algorithm has just two parameters.
- The clustering stage can be followed by one or two simplification stages:
    - Merging of spectrally similar classes;
    - Spatial smoothing.
- Good floater evaluation support is provided.
- A recalculable class scalar is produced in TSG.

Creating an FXClust scalar

Bring up TSG's scalar construction wizard, e.g., by using the Edit -> New scalar… menu.



You should see a new item **FXCLUST** in the Method list at the bottom of the first page. Select it and click Next.



No, come back – it's not that bad.

Clustering inputs

Features to extract

The main clustering inputs are wavelength centres of a sample's deepest 2 to 5 features. The feature extraction results that drive the clustering can be calculated by FXClust itself, or (for the SWIR) drawn from the dataset's hidden feature-extraction scalars. The following controls apply to feature selection.

'Empirical' checkbox

Turn this on to have the FXClust module do the basic feature-extraction calculations for each spectrum. If you have a TIR dataset then this checkbox is on and greyed out, meaning that the FXClust module will handle everything itself and you can't change it[1]. If you have a SWIR dataset then you may turn this checkbox off. If you do that, the FXClust will draw on the hidden feature-extraction scalars that TSG calculated when the dataset was imported[2].

Wavelength subset

TSG calculates feature-extraction results for each sample over the dataset's full wavelength range but you can narrow down the search for dominant features by specifying a wavelength subset[3].

The default subset is [2045, 2485] nm for Vis-SWIR and [6000, 14000] nm for TIR.

---

[1] FXClust has a more rudimentary way of doing feature extraction than the method used in TSG's import sequence. It drives clustering better in the TIR so you have no choice there.

[2] A reminder: Feature-extraction calculation settings can be changed in the FeatEx page of the File->settings property sheet. Changing these settings for a dataset will cause its hidden feature-extraction scalars to be updated.

[3] e.g., It is common to take just the "business end" of the SWIR – approx.. [2000, 2500] nm.

Enter your desired subset in the **Wvl** and **to** fields. If you like, you can enter these bounds in units other than the dataset's. Tell TSG your units in the accompanying **list**. Only features within the subset are considered.

Depth filtering

Although we are looking for the 2 to 5 deepest features for each spectrum, some spectra might be too weak to have *any* trustworthy feature-extraction results and you might be better off leaving them out of the clustering process altogether. They might just clutter the proceedings. You can use the **Min dep** field to specify a minimum acceptable feature depth.
If you have Empirical turned **on** (see above) then this threshold is in **reflectance space**. Examine some plots of reflectance spectra to get a feel for feature depth ranges.
If you have Empirical turned **off** then this threshold is in **feature-extraction space**. New to this build of TSG is **feature reporting in the Spectrum screen**. If you turn on the feature-tag annotation (Feats checkbox) and just have the main spectrum plotted (no second, reflib or aux spectrum) then you will get readouts of feature centres, depths & widths in the statusbar and at-cursor text. Browse some spectra and their features to get a feel for depth ranges.

Number of features

The **Features** list offers the choice of 2 to 5 features. E.g., If you select **3** then the wavelength centres of the 3 deepest features of each spectrum will be the main clustering inputs.
The default setting of **2** is normally adequate for a fair clustering classification.

Additional "control" inputs

You can supplement the feature-wavelength clustering inputs with two ***experimental*** variables. Each is controlled by a weight ranging 0 to 1. A control input is included if its weight is set above 0.

Depth

Given that we have found the two deepest features for a spectrum, the **Depth** control is based on the difference between the two feature depths.
It's arm-waving time. Sorry, see. Suppose we have a mixed zone where the spectra's deepest two features are very close in depth. For some spectra in the mixed zone the 2350nm feature is dominant while for others it's the 2200nm one. Dominance is much clearer in other less-mixed parts of the drillhole. Suppose the clustering algorithm finds one cluster for the [2350, 2200] group and a second cluster for the [2200, 2350] group (as one would hope). A mixed sample could find its way into either one of these very different clusters – the choice is whimsical. The cluster-merging stage of the process (based on spectral similarity and done after clustering) *might* merge the two clusters but it would then discard the separation we had for the less-mixed samples. Now if we include a Depth control input, the clustering algorithm will see mixed samples as "looking" different to the less-mixed ones. If luck (and the document fairy) is on our side we'll end up with 4 clusters here: [2350, 2200] mixed, [2350, 2200] normal, [2200, 2350] mixed, [2200, 2350] normal. It is likely that the two mixed clusters would be merged in the subsequent merge stage on account of their spectral similarity.
The Depth control is still available when 3 or more features are selected. It is still derived from the most dominant 2 features.

Spatial

The **Spatial** control is simply based on a sample's index in the dataset.

The clustering algorithm is usually oblivious to sample location. If a sample near the top of the drillhole has similar dominant features to one near the bottom then the two samples will probably end up in the same cluster regardless of their spatial separation. There is good sense in this. But consider this disingenuously contrived situation: "Judging by the cluster-mean spectra I'd say that cluster 5 is kaolinite but it's at both ends of the drillhole and it offends me to see 'kaolinite' at depth."
The Spatial control input will make the algorithm sensitive to sample location. A cluster that showed right through the drillhole is likely to get split into 2 or more clusters. This might yield an interesting result when combined with the cluster merging stage.

Mask

TSG's calculated scalars can all be filtered by an output mask but in the case of the FXClust scalar ***this is also an input mask*** and the selection list is labelled **Input / output mask** here as a reminder. Samples that are off in the mask will not drive the clustering.

Clustering algorithm parameters

TSG implements the original DBSCAN algorithm due to Ester et al[4]. There are two parameters and a checkbox.
The algorithm works with the concept of a "core point", for which there are at least "**min density**" other points within a distance of "**epsilon**". In the algorithm's outer loop through so-far-unclassified samples, a point that is found to be a "core point" starts a new cluster and engages the inner loop. In the inner loop, the current cluster is fleshed out from the core point. A point found within "epsilon" extends the current cluster, and if it is also found to be a "core point" then it can chain to further extension. So the algorithm can "feel out" shaped clusters (elongated, curved etc), not just blobs, and seems just the ticket when you look at some feature-extraction scatterplots.
The algorithm always allows for a noise class. Any sample that isn't a core point or within "epsilon" of a core point is left in the "noise" class.

Epsilon

The clustering algorithm works on feature-*wavelength* inputs (perhaps with similarly-presented control inputs). Epsilon is a distance measure – Euclidian distance in TSG, and in the same "space" as the feature wavelengths. It is in the same units as the ones you select for the wvl fields.

Min. density (points)

This one's a sample count. Think of this as kind of "critical mass" rather than a "cluster size".

Keep 'noise' class

The algorithm will always present a 'noise' class. Turn this checkbox off to discard it, and then any samples that were classified as 'noise' will be NULL.

---

[4] "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", "Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu", KDD-96 Proceedings. Available here: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf

Cluster merging pass

The primary clustering pass can be followed by a simplification pass, where clusters are combined if their *mean spectra* are sufficiently similar. Turn on the **Cluster merging pass** checkbox to enable this stage.

Min. correlation

The similarity measure here is Pearson's correlation, which ranges [-1, 1]. Adjust the lower cutoff as necessary. If two clusters' mean spectra have this high a correlation (or higher) then the two clusters will be merged. Hint: you probably want a cutoff *very close* to 1.
The "Means" floater evaluation plot and accompanying correlation-matrix display in the evaluation control panel can assist you in finding a suitable cutoff.

[Layer selection list]

The merging pass works on cluster mean spectra, and you can have these means computed from any spectral layer. Select the layer you want with the list.
For example, if you select the NormRefl layer then the cluster-mean spectra will be computed from normalised reflectance inputs. This is *not* the same thing as computing the means from reflectance inputs and then normalising the means[5].

Spatial smoothing pass

The procedure can be wrapped up with a final stage – spatial smoothing. Turn on the **Spatial smoothing pass** checkbox to enable this stage. It is available regardless of whether or not Cluster merging pass is enabled.
Spatial smoothing works in the full-dataset space (not the subset that's active in clustering), so is able to take a sample that was NULL[6] and assign it to a cluster. It works with a local histogram derived from a moving window. For example if you specify a window of size of 51 then a histogram (counts of samples per cluster) is calculated from the 51 samples around the current sample. NULL and noise-class samples are ignored. The most frequent cluster is found in the histogram and if it is different to the current sample's classification then the current sample *may* be reclassified accordingly. Its spectrum is checked against the proposed cluster's mean spectrum and if it is similar enough then the current sample *is* reclassified.

Window

This is the size, in samples, of the smoothing window described above.

Min. correlation

This is your sanity control. A sample can only be reclassified if the correlation between its spectrum and the proposed cluster's mean spectrum is this high (or higher).

[Layer selection list]

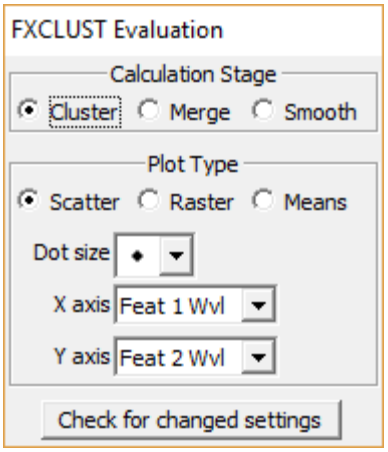
The spatial smoothing pass shares layer selection with the cluster merging pass. The layer selection list will be visible even if cluster merging is turned off. Layer selection affects the correlation sanity check.

---

[5] An aside: Pearson's correlation is driven by *shape* difference and it should not be affected by simple spectral albedo. Therefore you could be forgiven for thinking that the Reflectance and NormRefl layers should give the same correlations. It *is* possible (that you would be forgiven), but the correlations will probably be different.
[6] At this point a sample will be NULL if it was off in the mask *or* if it did not have viable features for clustering. If it was off in the mask then it will get turned off again during the standard output-masking step right at the end.

Evaluation plots

Click the **Evaluate** button in the scalar wizard for evaluation plots. **Floater 4** is taken over if it is up already and brought up if not.



There's quite a decent little selection of evaluation plots – more than any other TSG scalar has – and the Floater gets accompanied by a miniature **control panel** to handle the options. Before we get stuck in here, cast your gaze to the control-panel screengrab on the right and observe the button labelled **Check for changed settings**. A TSG Floater has a measure of independence and this evaluation floater is not under the leash of the scalar wizard. If you change a setting in the scalar wizard, the floater will not "see" it automatically. Click the Check for changed settings button if you change a setting. This will cause the floater to update its calculations as necessary.

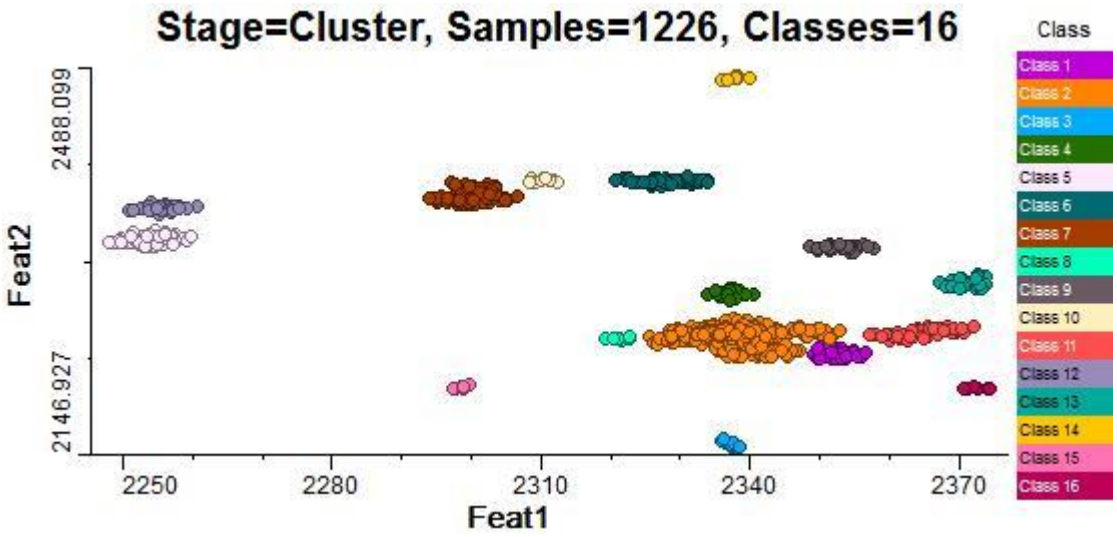
Calculation stage

You can flick through each of the three calculation stages easily – see how the Merge stage changed the basic clustering and how the Smooth stage tidied things up. If a stage is not enabled in the scalar wizard then it will look like the one before, e.g., if the Merge stage is turned off then its plots will be the same as the Cluster stage's.

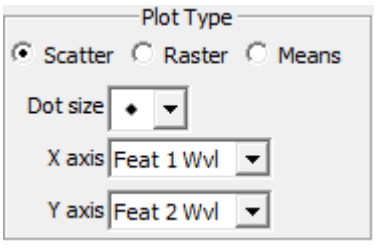
Plot type

Three different plots are available.

Scatter



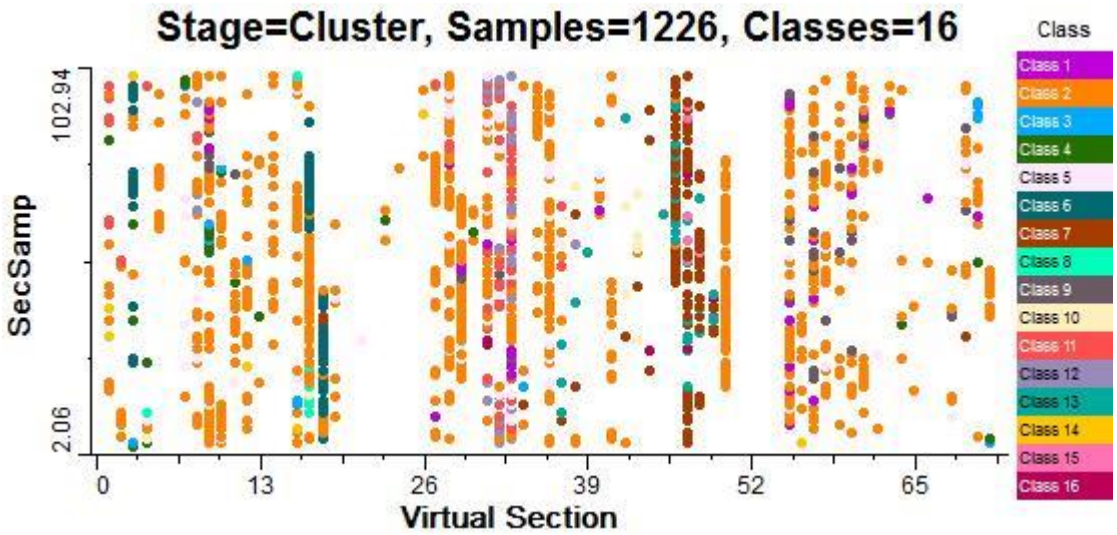
This is a scatterplot with two clustering input variables driving X and Y, and the classification driving colour. The default arrangement has the deepest feature's wavelength on X and the second-deepest's on Y, but you can select any of the inputs for X and Y using the controls. You can also adjust the dot size.



This plot is good at showing clusters in the context of the main drivers, but does not show point density. (There could be a multitude of superimposed points and you wouldn't know it from this plot.)

An aside: The scatter plot might look busier for the Smooth stage than the Cluster or Merge stage. This is because spatial smoothing can cause "weak" points ("noise" samples, or samples that didn't cut it as clustering inputs) to be added to clusters. The Smooth stage's correlation threshold can provide a sanity control here.
Click to get a cross-hair on the nearest plotted point *and navigate TSG main screen & other floaters to the sample concerned*.

Raster

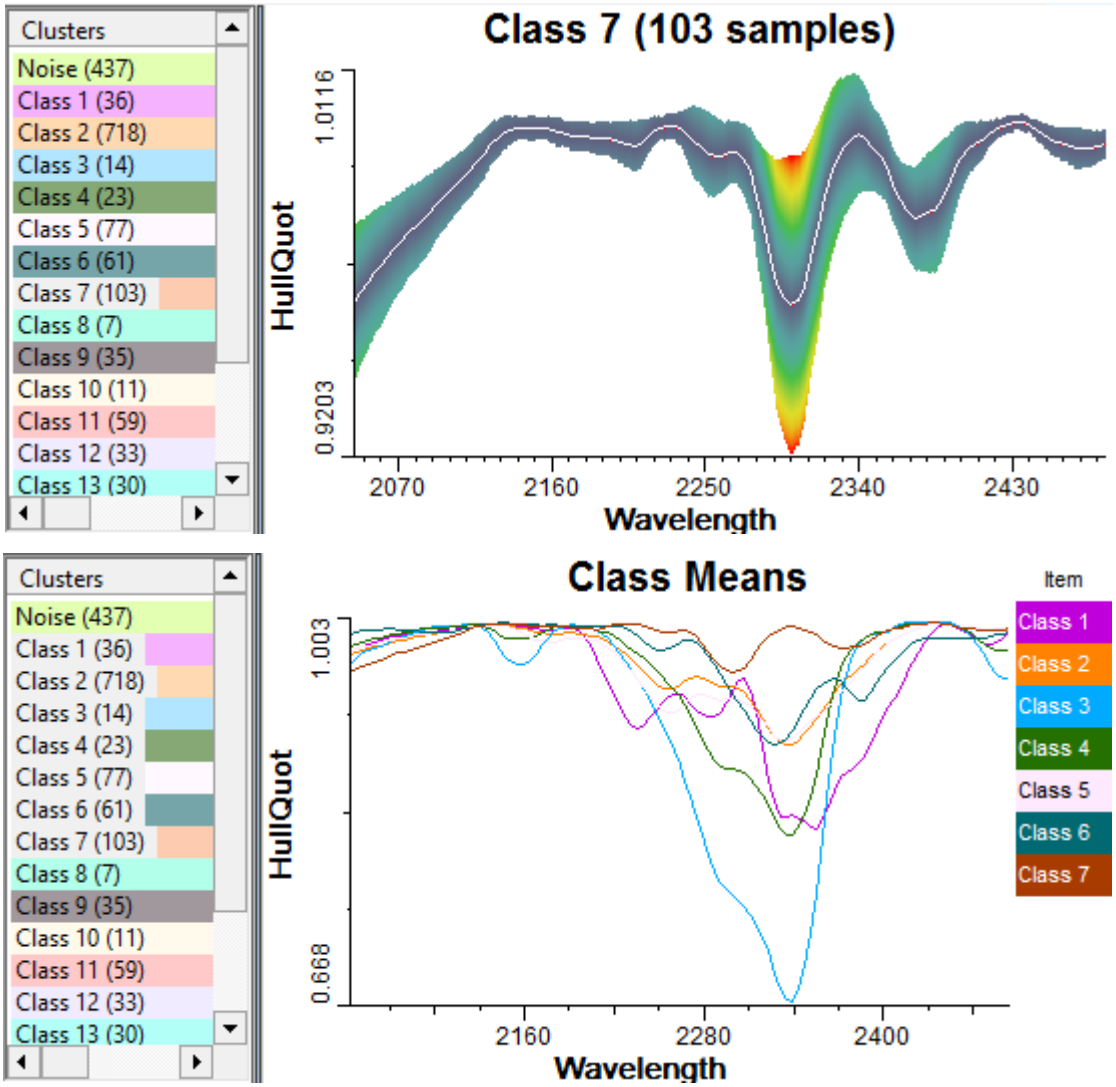


This isn't really a true raster plot but rather a rectangular plot rendered by TSG's scatterplot engine. It is driven by the HyLogging **Virtual Section** scalar on X, **SecSamp** scalar on Y, and sample classification driving colour. You can change the dot size using the list.

It presents the dataset spatially as one enormous core tray, with core sections going across and sample-in-section going up. It's a pretty useful plot.
It is only available for HyLogging-compatible datasets that have these scalars. SecSamp is an import-time scalar while Virtual Section is calculated. If you have a HyLogging dataset but aren't offered this plot then you might be missing Virtual Section. Go and (re)calculate[7] it ("core logging scalar" method), then return to clustering.
Cursor support is enabled and it can *navigate TSG's main screen & other floaters*.

[7] Virtual Section might be incorrectly marked up in old datasets. Recalculating it with the current TSG8 should sort it out.

Means



This plot shows one or more cluster-mean spectra. If you select just one in the floater list then you get the top plot with a fancy plus/minus 1-standard-deviation envelope, and you can see where (in the spectrum) the variability hot-spots are. If you select more than one then you get a straight overlay plot. (Select more than one by using<SHFT>click pairs or <CTRL>click on the list.)

The cluster mean spectra are calculated from the spectral layer selected in the scalar wizard. The mouse cursor works on the plot but doesn't trigger any navigation in TSG.

If there are less than 65 clusters then you get a square of coloured blocks in the control panel. It's a correlation matrix but in this case I'd prefer to call it a "matrix of correlations" because it isn't the usual correlation matrix in the spectral channel space. Each cell is the correlation between one cluster's mean spectrum and another cluster's mean spectrum. Red is high and blue is low. The diagonal elements are all red because they are 1. (Each diagonal element is a cluster mean spectrum's correlation with itself.) You can **roam the mouse around the matrix** to inspect correlations. The readout to the left of the matrix gives the cluster indices and Pearson's correlation for the little block under the mouse pointer. (Index 0 is the noise class.) You can **left-click a little block** to get a plot of the two cluster mean spectra concerned.